

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A switch for transferring data comprising:

at least one master unit;

a plurality of slave units;

a bus through which the master unit communicates with the slave units;

a detecting mechanism for detecting whether a slave unit is a failed slave unit, wherein if the slave unit is a failed slave unit, causing the switch to abnormally terminate;

and

memory in the switch comprising a software program that, upon abnormal termination of the switch, directs the at least one master unit to avoid further accessing a failed slave unit of the plurality of slave units by storing status information relating to the failed slave unit, in persistent storage, that survives across the abnormal termination of the switch and thereafter preventing the master unit from attempting to access the failed slave unit.

~~a memory in communication with the master unit having a software program that causes the master unit to automatically recover and restart when a slave unit fails, because the slave unit is non-operational, which has caused the master unit to fail and to avoid further accessing the failed slave unit after restart, and separately when there is a loss of power.~~

2. – 3. (Currently Amended)

4. (Currently Amended) The [[A]] switch as described in Claim 1 [[3]] wherein the software program causes the master unit to automatically recover when the detecting mechanism causes the master unit to abnormally terminate.

5. (Currently Amended) The ~~[[A]]~~ switch as described in Claim 4 wherein the detecting mechanism includes a hardware watchdog device.

6. (Currently Amended) A method for transferring data comprising the steps of:
attempting to access a ~~failed~~ slave unit of a plurality of slave units of a switch by a master unit of the switch with a signal through a bus through which the master unit and the ~~failed~~ slave unit communicate;

detecting whether the slave unit is a failed slave unit, wherein if said slave unit is a failed slave unit, causing the switch to abnormally terminate; and
responsive to abnormal termination of the switch, automatically recovering the switch from the failed slave unit and restarting the master unit which has failed because the failed slave unit failed and caused the master unit to fail with a software program in the switch that directs the master unit to avoid further accessing the failed slave unit of the plurality of slave units by storing status information relating to the failed slave unit in persistent storage that survives across abnormal termination of the switch and thereafter preventing the master unit from attempting to access the failed slave unit.

7. (Currently Amended) The ~~[[A]]~~ method as described in Claim 6 wherein the recovering step includes the step of obtaining status information about the slave units from said persistent storage.

8. – 9. (Canceled)

10. (Currently Amended) The method ~~A-program~~ as described in Claim 7 ~~[[8]]~~ including the step of changing information in persistent storage associated with the first slave unit from identified as failed to identified as good if the master unit does not terminate abnormally after the master unit attempts to contact the slave unit.

11. (Currently Amended) The method ~~A-program~~ as described in Claim 10 including the step of setting a variable slot chosen from amongst a plurality of slots of the switch not marked as potentially bad.
12. (Currently Amended) The method ~~A-program~~ as described in Claim 11 including the step of determining whether the first slave unit is physically present in a first slot of the plurality of slots.
13. (Currently Amended) The method ~~A-program~~ as described in Claim 12 including the step of determining the first slot is marked to be skipped.
14. (Currently Amended) The method ~~A-program~~ as described in Claim 13 including the step of marking the variable slot as potentially bad if it is not marked potentially bad.
15. (Currently Amended) The method ~~A-program~~ as described in Claim 14 including the step of reporting the variable slot as containing broken hardware and preventing the master unit from attempting to access the variable slot if the variable slot is marked to be skipped.
16. (Currently Amended) The method ~~A-program~~ as described in Claim 15 including the step of attempting to access hardware present in the variable slot if the variable slot is marked potentially bad.
17. (Currently Amended) The method ~~A-program~~ as described in Claim 16 including the step of marking the variable slot as good if the master unit did not abnormally terminate when the master unit accessed the first slave unit.
18. (Currently Amended) The method ~~A-program~~ as described in Claim 17 including the step of enabling normal operations on hardware present in the variable slot if the variable slot is marked as good.

19. (Currently Amended) The method A-program as described in Claim 18 including the step of setting the variable slot to a next slot of the plurality of slots.